RELEASE NOTES FOR MODELS-3 Version 4.1 Patch: SMOKE Tool and File Converter

CONTRACT NO. 68-W-99-002
TASK ORDER NO. 010

Prepared for:

United States Environmental Protection Agency
Office of Research and Development
National Exposure Research Laboratory
Mail Drop 80
Research Triangle Park, NC 27711

Task Order Project Officer:

William G. Benjey

Prepared by:

Systems Development Center
Science Applications International Corporation
6565 Arlington Boulevard
Falls Church, VA 22042

NOTICE

This publication is a part of the documentation the Models-3 Third Generation Air Quality Modeling System. The U.S. Environmental Protection Agency through its Office of Research and Development partially funded and collaborated in the procedures described here under contract 68-W-99-002 to Science Applications International Corporation. This document has been subjected to the Agency's peer and administrative review and has been approved for publication as an EPA document. Mention of trade names or commercial products is not intended to constitute endorsement or recommendation for use. The SMOKE Tool and File Converter were developed for release with Version 4.1 of the Models-3 air quality modeling system under contract to the U.S. Environmental Protection Agency and is free of restrictions to users. Users are requested to provide copies of derivative works to the government without restrictions as to use by others. Users are responsible for acquiring their own copies of commercial software associated with the SMOKE Tool portion of Models-3 release and are also responsible to those vendors for complying with any of the vendors' copyright and licence restrictions. Portions of the I/O API and SMOKE used with SMOKE Tool, and the Models-3 model builder are Copyrighted *1993-2001* by MCNC- North Carolina Supercomputing Center, and are used with their permissions subject to the above restrictions. Rights and restrictions for copyrights and trademarks used throughout this document are listed in *Appendix E* of the Models-3 Version 4.1 Installation Manual. Although efforts were made to make Models-3 user friendly, it is a complex software system. Consequently, it is important that the system's administrator read the Models-3 Installation Manual before installing the system, and this document before updating Models-3, including the SMOKE Tool. Installation instructions must be followed closely. There are error conditions which can come from third-party software, and are not Models-3 problems specifically. For example, expired or unavailable SAS or Arc/Info licences or versions not tested with Models-3 can result in error messages.

ABSTRACT

This software patch to the Models-3 system corrects minor errors in the Models-3 framework, provides substantial improvements in the ASCII to I/O API format conversion of the File Converter utility, and new functionalities for the SMOKE Tool. Version 4.1 of the Models-3 system must be installed before applying this patch. The SMOKE Tool is an emission modeling component of Models-3, a flexible system designed to simplify the development and use of air quality models and other environmental decision support tools. Models-3 is designed for applications ranging from regulatory and policy analysis to understanding the complex interactions of atmospheric chemistry and physics. This document describes corrections to the Models-3 framework system, describes the procedures used by SMOKE Tool to grid spatial surrogate data, describes how additional spatial surrogate coverages may be

added, and describes a new reverse gridding capability (gridded data to county or other geographic boundary data).

FOREWORD

The Models-3 Community Multi-scale Air Quality Model (CMAQ) modeling system has been developed and improved under the leadership of the Atmospheric Modeling Division of the EPA National Exposure Research Laboratory in Research Triangle Park, NC. The SMOKE Tool along with the SMOKE emission modeling system, is an important part of the summer 2001 Sun UNIX and NT-based releases (Version 4.1) of Models-3. This document and related software patch provide minor corrections to Version 4.1 and substantial new functionality for the SMOKE Tool. Other than third-party costs as described in the Notice, the SMOKE Tool portion of Models-3 is available without charge for use by air-quality regulators, policy makers, industry, and scientists to address multi-scale, multi-pollutant air quality concerns.

In keeping with its capabilities Models-3 , including the SMOKE Tool, is a sophisticated and complex system.. Consequently, before applying this patch, the system administrator is advised to read the Models-3 Version 4.1 System Installation and Operation Manual and SMOKE Tool for Models-3 Version 4.1 document, as well as this report. The SMOKE Tool, in concert with SMOKE, represents another technical aid in a the process of improving Models-3 capabilities in and allowing a truly multi-pollutant approach to air quality studies.

William B. Petersen
September 2001

CONTENTS

EXHIBITS

1.0 INTRODUCTION

The U.S. Environmental Protection Agency (EPA) is empowered by current legislation to address today's important, grand challenge environmental problems, such as air quality management and linked air and water quality management. The use of environmental models is essential to EPA's cost effective accomplishment of its mission. As part of its High Performance Computing and Communications (HPCC) program, the Agency has initiated the development of a third-generation air quality modeling system called Models-3. Models-3 directly supports the HPCC program goal of using emerging computing and communications technology to:

- Develop the Agency's capability to perform complex multi-pollutant and cross-media pollutant studies that are currently not feasible due to computational limitations.

- Build federal, state, and industrial capabilities to use advanced assessment tools directly responsive to the needs for environmental management.

- Position the Agency to more easily integrate emerging computing and communications technology into environmental assessment tools, ensuring the most reliable and timely response to key environmental issues.

The Models-3 project is one of several developed under the HPCC program. In many respects, Models-3 has been the primary focus of EPA's HPCC research activities in that all the other projects under HPCC are, in one way or another, going to make significant contributions to enhance the capabilities of the Models-3 system. The other projects include:
1) linkage of air and water quality models, 2) development and performance evaluation of numerical algorithms on parallel architectures for key environmental processes, 3) development of training and technology transfer approaches to provide state, federal, and industrial air quality scientists and decision makers with credible and useful air quality modeling and decision support tools, 4) implementation of advanced computing hardware, software, and network infrastructure to support HPCC research activities, 5) development of advanced collaboratory visualization and analysis techniques, and 6) outreach activities designed to introduce more EPA researchers to the benefits of HPCC technology in their own areas of interest.

Legal Authority: The High Performance Computing Act of 1991 authorized EPA to perform research directed toward incorporating advances in computing and communications technology into EPA's environmental assessment applications and transferring these advanced tools to key state, federal, and industrial users with decision-making responsibility.

Regulatory Authority: As mandated by Congress in the 1990 Clean Air Act Amendments, the states are required to use photochemical grid models to develop credible modeling demonstrations for attaining the National Ambient Air Quality Standards (NAAQS) as part of their State Implementation Plans.

Regional Acid Deposition Model (RADM) applications are required to support EPA client and policy offices, EPA regional offices, and the states in implementing mandates of the 1990 Clean Air Act Amendments, special agreements between states, and treaty obligations under the U.S.–Canada Air Quality Agreement and the U.N. Economic Commission for Europe. The capabilities of RADM and ROM are improved and incorporated into the Community Multiscale Air Quality (CMAQ) model that is a part of the Models-3 Air Quality Modeling framework. CMAQ is a multi-pollutant model, enabling a unified "one-atmosphere" modeling approach and more realistic modeling of chemical interactions in meeting many of the air quality modeling needs of the 1990 Clean Air Act.

1.1 Purpose

This document provides new information not appearing in other Models-3 documentation and discusses changes since the last software release (Version 4.1). Anyone encountering a problem with Models-3 should check this document.

1.2 Scope

This document covers the Models-3 framework, the Sparse Matrix Operator Kernel Emission (SMOKE) Tool, and problems that may occur in getting to the individual models or tools used in the system. Problems with the actual models or tools are not covered in this document.

1.3 Identification

The sponsor of Models-3 is the National Exposure Research Laboratory (NERL) in EPA's Office of Research and Development. Work on the Version 4.1 Patch of Models-3 was performed under Mission Oriented Systems Engineering Support (MOSES) Task Order 002-010. The System Development Center (SDC) product control number for this patch release is SDC-0002-010-JS-3010.

2.0 REFERENCES

Byun, D.W. and J.K.S. Ching. *Science Algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System*. EPA/600/R-99/030, National Exposure Research Laboratory, Research Triangle Park, NC, 768 pp. (1999).

Novak, J.H., R.L. Dennis, D.W. Byun, J.E. Pleim, K.J. Galluppi, C.J. Coats, S. Chall, & M.A. Vouk. *EPA Third Generation Air Quality Modeling System, Models-3 Volume 1: Concept*. EPA/600/R95/084, National Exposure Research Laboratory, Research Triangle Park, NC, 55 pp. (1995).

System Installation and Operation Manual for the EPA Third-Generation Air Quality Modeling System (Models-3 Version 4.1 for Sun Unix and NT), National Exposure Laboratory Research Laboratory, Research Triangle Park, NC, (2001).

EPA Third Generation Air Quality Modeling System, Models-3 Volume 9b: User Manual. EPA/600/R-98/069(b), National Exposure Research Laboratory, Research Triangle Park, NC, 833 pp. (1998) (To Be Updated).

EPA Third Generation Air Quality Modeling System, Models-3 Volume 9c: User Manual: Standard Tutorial. EPA/600/R-98/069(c), National Exposure Research Laboratory, Research Triangle Park, NC, 387 pp. (1998) (To Be Updated).

Revised Project Plan for Implementation Support, Maintenance, and Enhancement of the Third Generation Air Quality Modeling System (Models-3), SDC-0002-010-SR-3001A,
September 22, 2000.

Release Notes of Models-3 (Version 4.1), SDC-0002-SR-3005, April 3,2001.

3.0 SUMMARY OF FUNCTIONALITY

Models-3 is not a single model or modeling system, but rather a problem-solving environment comprising components that help you build, evaluate, and apply air quality models. Each shaded rectangle shown on Exhibit 1 represents a component that helps you perform major functions associated with environmental modeling.

Models-3's Object Oriented Data Base Management System (OO DBMS) provides a centralized method for managing and sharing data, metadata (information about data), program and/or script executables, and relationships and dependencies among executables and data needed for a specific modeling study. Each of the components communicates with the OO DBMS as shown in Exhibit 1. These components serve as user-friendly tools for accessing or modifying datasets and program source code, developing program or script executables, performing model simulations, and visualizing and analyzing model simulation results.

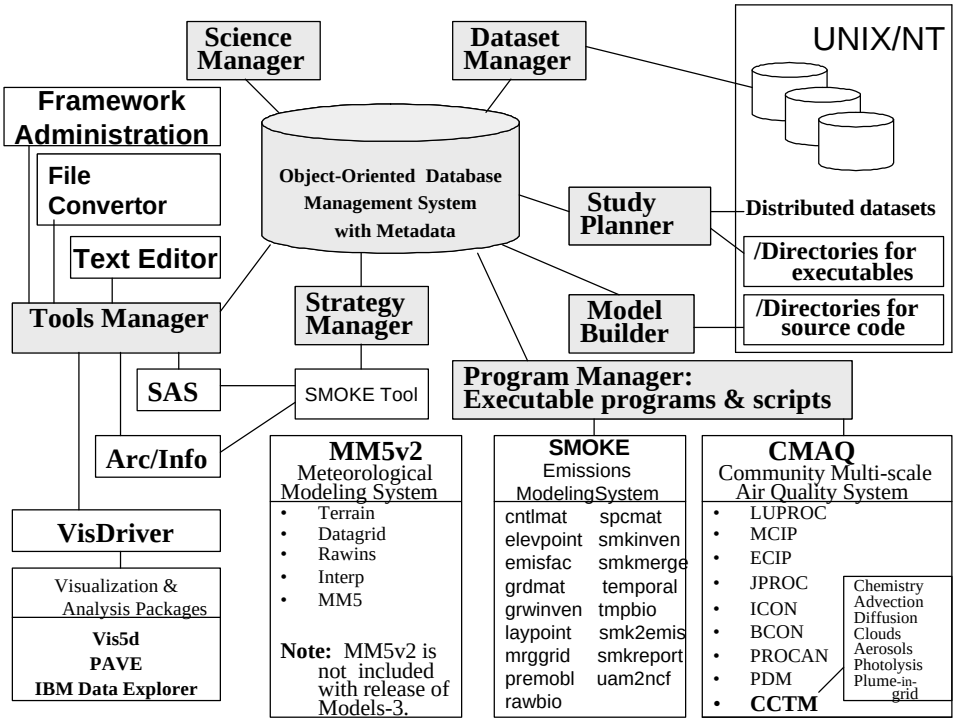

Exhibit 1. Models-3 Framework Components & Integrated Programs

The major Models-3 framework components are:

- **Science Manager** - allows you to define globally shared information on key science components.

- **Dataset Manager -** provides you with the capability to register datasets for use with modeling and analysis programs within Models-3.

- **Study Planner** - allows you to define a study and control the execution of its associated models and processors.

- **Model Builder** - allows you to prepare a model for a specific area for execution. You have the ability to select horizontal grid resolution, vertical layering resolution, and chemical mechanism, without the need for reprogramming.

- **Program Manager** - allows you to register, update, and search for executable programs and/or scripts to make them available for use in defining studies within the Study Planner component.

- **Strategy Manager** - allows you to interactively execute the SMOKE Tool. SMOKE Tool documentation is available via the Help icon at the Strategy Manager screen.

- **Tools Manager** - accesses a variety of administration, visualization, statistical analysis, and file conversion tools. The tools are Framework Administration (for the Models-3 Administrator), File Convertor, Text Editor, VisDriver (which provides access to Vis5 and Package for Analysis and Visualization of Environmental Data [PAVE], ArcInfo, and Statistical Analysis System (SAS).

4.0 DEGREE OF FUNCTIONALITY PROVIDED

This section provides a list of software changes since the last release, any assumptions or limitations of the Models-3 framework broken down by Models-3 component, and any problems found during testing of the released patch version.

4.1 Changes Since Last Release

The changes made in this Models-3 Version 4.1 Patch release were to the SMOKE Tool and the program used for Input/Output Application Programming Interface (I/O API) file conversions. No changes were made to the actual Models-3 Version 4.1 framework.

4.1.1 SMOKE Tool

A “Reverse Gridding” processor that converts gridded data to state or county level data was added to the SMOKE Tool. Documentation for the Reverse Gridding processor is included as Appendix A. Documentation for the SMOKE Tool was included in *Release Notes of Models-3 (Version 4.1)*, April 3,2001. There were also some “clean-up” changes made to the SMOKE Tool, therefore, an entire re-release of the SMOKE Tool is included with this patch release.

4.1.2 I/O API File Convertor

The I/O API File Convertor program has been changed so that it takes P_XCENT_GD and P_YCENT_GD for the x and y centers in the grid description file, instead of XCENT_GD and YCENT_GD. Additionally, a change was made such that when a grid description file has 0 (or null) layers, the number of layers is then set to 1.

The I/O API conversion program of the File Convertor can be used in a standalone mode outside of Models-3. It can also be registered as a program via the Models-3 Program Manager and then used as an executable node in study plans.

Documentation to use it as a standalone I/O API conversion program is included as Appendix B.

4.2 Assumptions, Limitations, and Problems

The following paragraphs present assumptions, limitations, and problems with various Models-3 components.

4.2.1 Presenter

This section addresses common problems found in the graphical interface of the Models-3 framework.

- If you have selected a remote host that has been registered as Orbix-enabled but Orbix is not running, problems can occur, and the Models-3 framework may crash.

- The print function will not work properly in certain complex windows. The problem appears in the Galaxy libraries. No immediate action is planned to remedy the problem.

- Every object (e.g., dataset, program, etc.) has a unique identification (ID) that you normally would not need to know. However, when uniqueness is required, the ID can assist you in distinguishing among different objects with the same name. You can reveal the IDs in both the summary and detail windows. Simply enlarge the summary windows and an ID column will appear. In the detail windows, go to the menu bar, click *View*, and select *Hidden Info*. An informational pop-up window that includes the ID will appear.

- If you exit while a save is in progress, an “object not saved” type of message appears. You should select *Cancel* and wait to avoid any problems. If the object has been saved, a “saved successfully” type of message is displayed. If you have saved an object but it has taken a while and no message has appeared, the object can be saved again.

- **User Hint:** Save frequently when adding or changing information. As an example, in Study Planner when annotating file links you may have many file links to annotate in a plan. A good rule of thumb is you should not annotate more than three links without doing a Save.

- On rare occasions, the Orbix daemon may get suspended when it is started directly from Unix. This is sometimes caused by the Concurrent Versions

System (CVS). You should always use the “m3run” script rather than from a binary executable to keep this from happening.

- When an object other than a dataset is copied, the server thinks it is editing the original object. If the original object is edited before the copy is saved, the message “Someone else is editing this object” is displayed. The message can be ignored. After saving the object, the message will not appear again.

- When using the NT or X-Window emulators, the system pauses and waits for you to do something. If the mouse is moved, the system will continue. This is a problem in the Galaxy library and cannot be fixed at this time. **It is important to occasionally move the mouse when long operations are occurring. This situation occurs frequently in the NT version when a node is being executed in Study Planner and it is time to update the screen.** The execution does not stop, but you will not know the current status of the execution until the mouse is moved.

- “Find” screens will accept more than 16 characters in a Name (Dataset, Study, Plan, Program, etc.) field. Creation screens limit the names to 16 characters. The user should never enter more than 16 characters for a Name on a find screen.

- On the NT, if Models-3 is not acting normally (e.g., you are having trouble doing a Save), there is a possibility that one of the Models-3 subsystems has aborted. Exit Models-3 when this happens. Next focus on the ORBIX window and do a <**Ctrl C**>. Then exit the two windows that were being used by Models-3, start up two new windows, and reinitiate Models-3.

4.2.2 Dataset Manager

- When selecting *Dataset/Visualize* at the menu bar of a Dataset Detail Window, a Vis5d format file is expected unless the file is in I/O API format. If the file format is
  I/O API, the VisDriver Program is displayed automatically. To use an I/O API format file, VisDriver from the Tools Manager can also be used.

- The *Dataset/Move* does not catch the error when you specify a file that is really the same file but different hosts were provided. (The file is mounted on both hosts.)

4.2.3 Study Planner

- On rare occasions when executing a plan, a warning message displays stating that “The Study Planner has experienced a problem and has restarted. It will no longer provide you with feedback ....” If this message occurs, you should click on the warning message *OK* button, select the Plan tab, then copy the selected plan (i.e., at menu bar click *Edit* > *Copy* > *Paste*), then select the New plan, click the Details icon, rename the New plan, and Save. Execution from the copy of the original plan should proceed without a problem. You should delete the original plan, and you can rename the New plan back to the original plan name.

- When copying selected nodes, datasets, or both with their connected edges from one plan to another, the edges are not correctly updated visibly on the screen when an individual node or dataset is moved. This is corrected after saving and resetting the study.

- When clearing an environment variable that is a higher-level environment variable (i.e., a study or plan level) from the Node Properties screen, the environment variable will not be displayed when going back into the Node Properties until some other action is done (e.g., pressing *OK* on the Node pop-up then coming back in).

- You cannot get a file from, or put one to, a non-Orbix-enabled machine without a “models3” user by using a data source or sink, when the execution of the node being used is done on a different (from the non-Orbix-enabled) machine. This is because Models-3 does not have a user ID and password for the non-Orbix machine. A dummy node can be created on the non-Orbix machine that allows the file to be specified as the output (or input, if using as a sink).

- You cannot copy a file between two non-Orbix-enabled machines if they do not use the same ID/password.

- Models-3 will abort if the user attempts to ‘Exit’ a plan when there is still an active Annotate Link Properties screen for the plan. The work-around is to ‘Exit’ all Annotate Link Properties screens displayed for a plan before exiting the plan detail screen.

- There are five places where the execution host can be set in Models-3 that can affect a plan execution -- in program registration, at a study detail window, at a plan detail window, and when annotating file links (both ‘To’ and ‘From’) at the Define Physical File Location screen. In the Execution Host pick list selection, a user can select [Default Host] from the list and the host name is displayed within square brackets. The user can also explicitly select the same host in the list

whereby the host is displayed without brackets. **Be consistent in the way you select the host (i.e., default or explicit)**. The reason for this is Models-3 Study Planner sometimes does not recognize that the host name in square brackets assigned in one place is the same as the same host name assigned without the square brackets. Being inconsistent can cause Study Planner to hang in execution, overwrite a file (Unix platform), attempt to write to the wrong drive (NT platform), or cause Models-3 to request the user to enter a password. An additional measure of protection against overwriting files should also be taken by making static input files (e.g., CEM data and inventory files) such that they do **not** have write permission.

- **User Hint**: In Study Planner, when annotating an input file link at the Link Information screen, there is a 'From' link and a 'To' link (below the 'From' link). If using the pick lists to select the files, select the 'To' pick list button before the 'From' pick list button to view the list of input logical file names specified for the program assigned to the node.

- **User Hint:** In Study Planner, when annotating physical file links, use the 'Physical File' pick list button then select 'Filter' at the Select File window. By using the filtering capability to establish a file path and name instead of typing them in, you eliminate the possibility of a typographical error.

4.2.4 Strategy Manager

There are no known problems.

4.2.5 Program Manager

- A physical file is not deleted when delete is selected. By design, only the metadata is deleted.

4.2.6 Science Manager

- If data is entered for the horizontal grid specifications that do not allow the grid to be viewed, an error message is displayed in the Orbix window.

- In the Vertical Layering Detail Window, you can generate the vertical layers as many times as you want. Once the metadata is saved, you cannot change the vertical layering type or regenerate the vertical layers. However, you can add to or otherwise modify the existing vertical layers.

- When editing two horizontal coordinate systems at the same time, only one projection parameters pop-up window can be displayed at a time.

- When copying a coordinate system at the Coordinate System detail page, after selecting a coordinate system to copy, you must first click on the *Projection* button and then *Component/Copy* at the menu bar. If you do not first click on the *Projection* button, the projection parameters will not be included in the copy.

- For GridViewer to work in the NT version, an X-Window server must be on the system, and the X-Window server must have been already initiated by the user. GridViewer is the program run when *View Grid* is selected on the Grid Definition tabbed page of the Horizontal Grid Manager.

4.2.7 Model Builder

- When you do a Model Build, the system issues a "The model has been generated" message when the build execution completes. The system does not determine, though, if errors were in the build itself. You should click the *View Build Log* and *View Compile Log* buttons after the message is issued to be sure that no errors are in the logs.

- **Note:** Model Builder is not implemented for the Cray computer.

4.2.8 Tools Manager

- In order to bring up VisDriver on NT, both Interix and an X-Window server must be on the system, and the X-Window server must have been already initiated by the user.

4.2.9 Execution Manager

There are no known problems.

4.2.10 File Migrator

- For File Migrator to function best, there should be a "models3" user on all Unix systems. Also, the M3TEMP directory should be a mounted directory if using a machine other than the one the servers are on.

- Remember that to function properly on Unix systems, the .rhosts file of each user of any non-ORBIX-enabled machine must allow access to both the user and the machine that is running the File Migrator. Please refer to the *System Installation and Operation Manual for the EPA Third-Generation Air Quality Modeling System (Models-3 Version 4.1 for Sun Unix and NT)* for more instructions.

4.2.11 Help

- Help is displayed through an Internet browser. Your system must have a default browser installed to see Help. Also, the browser must take a command line argument. If the browser is already started, the system will not "un-iconize" it and bring it to the front.

- *Index* and *Tutorial* from the Help pull-down menu are not implemented.

4.2.12 File Convertor

- When using free format and specifying a Fw.d format, the w (width) is ignored, and the display for a value less than 1 is 0.xx with d decimal accuracy.

- Free format using a blank delimiter has been changed to process multiple blanks as one delimiter.

- **Note:** File Convertor has the ability to do simple units conversions where the conversions are a multiplicative factor. By design it does not have the ability to do conversions where more than just a multiplicative factor is involved [e.g., C = 5/9 (F - 32)]. Therefore, in File Convertor at the Describe Variable screen, you will not find 'Temperature' in the Category pick list. File Convertor does not have the ability to convert between different temperature scales.

- The process of converting between SAS format and I/O API format files described in the Models-3 user manual no longer applies. The way to convert between SAS and I/O API files is a two step process. Do one run converting from I/O API (or SAS) to ASCII, and then do a second run converting from ASCII to SAS (or I/O API).

- I/O API files have a header containing temporal, gridding, and vertical layering information. To do an ASCII to I/O API or I/O API to I/O API file conversion, this information is entered via the Select Criteria screen accessed by clicking the *Selection* button at the File Convertor main screen.

5.0 INSTALLATION INSTRUCTIONS

Please refer to *System Installation and Operation Manual for the EPA Third-Generation Air Quality Modeling System (Models-3 Version 4.1 for Sun Unix and NT)* for installation of Models-3 Version 4.1.

APPENDIX A

SMOKE Tool Reverse Gridding Program Documentation

REVERSE GRIDDING PROGRAM (GRIDGEO): GRID TO GEOGRAPHIC AREA

Program gridgeo converts gridded data to geographic area data using spatial surrogates generated by the SMOKE Tool spatial surrogate processor. The output files may either be
I/O API or ASCII. Both types may be specified in the same run. The environment variables that specify the computation options are:

LEVEL	Output file level. Number of characters to use from the area Ids on the surrogate file. Only entered if the geographic areas desired are at a higher level than those defined on the surrogate file. Defaults to the length of the ID on the first surrogate record.	Optional
DEBUG	Option to generate debug report. Valid values are YES, Y, NO, and N. Values are case-insensitive. Default is NO.	Optional

The environment variables that specify input files are:

GRIDDED	Gridded I/O API input file. File must contain 2-d or 3-d gridded data.	Required
SURROGATES	Spatial surrogate file. Contains geographic area spatial surrogates. Generated by the SMOKE Tool spatial surrogate processor from a non-county universal coverage.	Required
VAR_TABLE	Variable table. Defines the computation method for variables (e.g., DOMINANT, AVERAGE, or TOTAL).	Optional

The environment variables that specify output files are:

OUT_API	Name of I/O API output file.	Optional
OUT_APIX	Name of I/O API output file index. Contains state or county codes for I/O API output file. (ASCII file)	Optional
OUT_ASCII	Name of ASCII output file.	Optional
REPORT	Name of debug report file.	Optional
LOGFILE	Name of I/O API log file.	Optional

The program will only generate the output files for environment variables that have been specified. For example, if OUT_ASCII is not defined, the ASCII output file will not be generated. Both OUT_API and OUT_APIX must be specified to generate an I/O API output file. Messages are written to the standard output (Unix or NT logs).

Variable Table

The variable table specifies the way that variables will be converted to geographic area data. The table is a free-format space-delimited ASCII file that contains the following fields on each record:

Name	Description	
NAME	Variable name.	Required
METHOD	Computation method (DOMINANT, AVERAGE or TOTAL). Uses default computation method for the variable type, if missing. Case insensitive.	Required

All input file variables will be written to the output file. Variables not defined in the variable table will use the default computation method for the variable type.

The computation methods are:

DOMINANT - Finds the dominant value (or code) for the state/county. DOMINANT can only be used for integer variables. The DOMINANT method uses the value that has the largest surrogate value in the geographic area.

AVERAGE - Computes the average value for a geographic area from the contributing cells.

TOTAL - Computes the total cell contributions to the geographic area. SUM may be used as a synonym for TOTAL.

The example below illustrates the DOMINANT methodology. The following table shows the spatial surrogates defined for a geographic area, along with the value of each cell in the array being converted. The surrogate value is the value for the area-cell intersection.

ID	Cell	Surrogate Value	Cell Value
026017	3,3	1000	2

026017	3,4	3000	3
026017	4,3	1000	2

The dominant value for area 026017 is 3. The surrogate totals computed for each cell value are:

Cell Value	Surrogate Total
2	2000
3	3000

The equations used for AVERAGE and TOTAL are:

$$\sum_i \left(value_i \times srgval_i \right) \div \sum_i srgval_i \qquad \text{(Average)}$$

$$\sum_i \left(value_i \times srgval_i \div celltot_i \right) \qquad \text{(Total)}$$

The subscript i used in the equations represents the ith cell for which spatial surrogates are defined in the geographic area. The variables used in the equations are:

$value_i$ Cell value.
$srgval_i$ Surrogate value for area-cell intersection.
$celltot_i$ Total surrogate value for cell. Sum of all area-cell intersections for cell.

The default computational method for each variable type are:

Variable Type	Default Computational Method
INTEGER	DOMINANT
REAL	AVERAGE
DOUBLE	AVERAGE

ASCII Output File

The ASCII output file is a free-form space-delimited file. It consists of a set of header records followed by data records. The header records are:

#GRID grid description (one record)
#CASE case description (one record, if input file time-stepped)
#VAR variable description (one record for each variable on output records)

The grid header record is in the same format as the grid header record on the spatial surrogate file. The parameters are:

Parameter	Name	Type	Description
1	GDNAME	Char	Grid name (up to 16 characters)
2	XORIG	Real	X origin (projection units)
3	YORIG	Real	Y origin (projection units)
4	XCELL	Real	Cell size, X direction (projection units)
5	YCELL	Real	Cell size, Y direction (projection units)
6	NCOLS	Int	Number of cells in X direction
7	NROWS	Int	Number of cells in Y direction
8	NTHIK	Int	Boundary thickness (number of cells)
9	PROJTYPE	Char	Project type (UTM, LAMBERT, LAT-LON,...)
10	PROJUNIT	Char	Projection units (DEGREES or METERS)
11	P_ALP	Real	First map projection description parameter
12	P_BET	Real	Second map projection description parameter
13	P_GAM	Real	Third map projection description parameter
14	P_XCENT	Real	Center of coordinate system (degrees longitude)
15	P_YCENT	Real	Center of coordinate system (degrees latitude)

The case header record is only included when the file is time-stepped. The parameters are:

Parameter	Name	Type	Description
1	DATEA	Date	Start date (yyyyddd)
2	TIMEA	Time	Start time (hhmmss)
3	TSTEP	Time	Time Step increment (hhmmss)
4	NSTEP	Int	Number of time steps

The variable header records define the variables on each data record. There is one record for each variable. The parameters are:

Parameter	Name	Type	Description
1	POS	Int	Position on data record
2	NAME	Char	Name
3	TYPE	Char	Type
4	UNITS	Char	Units
5	DESC	Char	Description

The remainder of the file contains the data records. Each data record contains the variable data for an array entry. The initial variables on a data record contains the record key that defines the entry. The key variables that can be on a record are:

Name	Type	Description	
DATE	Date	Time step date (yyyyddd)	Only if file time-stepped
TIME	Time	Time step time (hhmmss)	Only if file time-stepped
LAYER	Int	Layer number	Only if file has multiple layers
ID	Char	Geographic area ID	

The remaining variables on the record contain the data for the entry. These variables will be type INT, REAL, or DBLE.

Debug Report

The debug report contains the general messages printed to the system log and the reverse gridding factors used for each variable. The general messages describe the grid, give the names of input and output files, describe the manner in which variables will be converted, and give summary counts for the output files.

The following report lines show the factors used for a county whose computation method is AVERAGE:

```
URW_FLAG 026063   2  12  srgval,cytot= 8.79623E+07 8.79623E+07
URW_FLAG 026063   3  12  srgval,cytot= 3.03179E+08 3.91141E+08
URW_FLAG 026063   4  12  srgval,cytot= 2.70276E+08 6.61417E+08
```

URW_FLAG 026063 5 12 srgval,cytot= 7.47941E+07 7.36211E+08

The lines contain the following elements:

The variable name (URW_FLAG).

The geographic area ID (026063).

The column and row numbers for the grid cell.

The surrogate value for the area-grid cell intersection (srgval).

The total surrogate contribution to the geographic area from grid cells(cytot). The report lines show a running total from the individual cells. The final value for the county is the number used in the calculation.

The following report lines show the factors used for a county whose computation method is TOTAL:

USGS_TYPE 026063 2 12 srgval,celltot,ratio= 8.79623E+07 3.59210E+08 0.24488
USGS_TYPE 026063 3 12 srgval,celltot,ratio= 3.03179E+08 6.25000E+08 0.48509
USGS_TYPE 026063 4 12 srgval,celltot,ratio= 2.70276E+08 6.25000E+08 0.43244
USGS_TYPE 026063 5 12 srgval,celltot,ratio= 7.47941E+07 2.09261E+08 0.35742

The lines contain the following elements:

The variable name (USGS_TYPE).

The geographic area ID (026063).

The column and row numbers for the grid cell.

The surrogate value for the area-cell intersection (srgval).

The total surrogate value for the grid cell (celltot).

The fraction of the surrogate for the grid cell in the county (ratio).

Program Execution

The program can be executed with the following commands:

$M3MSTOOL/bin/execProgram gridgeo (On Unix)
%M3MSTOOL%\bin\execProgram gridgeo (On NT)

The Unix form of the command should be used in a Models-3 program definition. Then, the same program can be used for both Unix and NT plans.

The script is designed to run under Models-3. If running as a standalone script (outside of Models-3), the following environment variables will also have to be set.

STUDY_PATH	Name of directory where the program log and list files will be stored.
M3FERRORFILE	Step status file. Gives completion status to Models-3, only on NT.

The script automatically assigns environment variables LOGFILE and REPORT to files $STUDY_PATH/program.log and $STUDY_PATH/program.lst, respectively. These are the program log and report files for a study planner node.

The Unix script selects the correct gridgeo for the platform using the current operating system name from the uname command. The SMOKETOOL directories contain a subdirectory (under bin) for each platform supported. Currently the only Unix platform supported is Sun (subdirectory **SunOS**). The NT subdirectory is **win**.

Sample statements for running a conversion on the Sun are shown below:

```
#!/bin/csh -f
setenv  STUDY_PATH  /home/susick/test/GRIDGEO

setenv  DEBUG       yes
setenv  VAR_TABLE   $STUDY_PATH/lake_erie.vartab
setenv  GRIDDED     /home/susick/BELD3_LANDUSE/dynamic_nt/lake_erie_totals.ioapi
setenv  SURROGATES  /m3aux/emis/susick/smoketool/gridspec/lake_erie/sas/cy_ratios.dat
setenv  OUT_API     $STUDY_PATH/lake_eriex.ioapi
setenv  OUT_APIX    $STUDY_PATH/lake_eriex.ioapix
setenv  OUT_ASCII   $STUDY_PATH/lake_eriex.dat

rm  $OUT_API
rm  $OUT_APIX
$M3MSTOOL/bin/execProgram  gridgeo
```

Sample statements for running a conversion on the NT are shown below:

```
set  BASE=x:\test\GRIDGEO
set  STUDY_PATH=x:\test\GRIDGEO\NT
set  DEBUG=yes
set  VAR_TABLE=%BASE%\lake_erie.vartab
set  GRIDDED=x:\BELD3_LANDUSE\dynamic_nt\lake_erie_totals.ioapi
set  SURROGATES=Z:\emis\susick\smoketool\gridspec\lake_erie\sas\cy_ratios.dat
```

```
set  OUT_API=%STUDY_PATH%\lake_erie.ioapi
set  OUT_APIX=%STUDY_PATH%\lake_erie.ioapix
set  OUT_ASCII=%STUDY_PATH%\lake_erie.dat
set  M3FERRORFILE=%STUDY_PATH%\lake_erie.status
del  %OUT_API%
del  %OUT_APIX%
%M3MSTOOL%\bin\execProgram gridgeo
```

Both sets of statements execute the same run, which:

Generate I/O API and ASCII output files at the geographic level of the spatial surrogate file.

Uses a variable table with the following records:

- URW_FLAG avg
- USGS_TYPE sum

Generates a debug report.

A sample debug report is shown below. To save space, most of the conversion factor lines have been dropped. "---- lines dropped ----" messages have been included where the lines were dropped from the report.

```
 ---- Output File  8, /home/susick/test/GRIDGEO/program.lst
 GRIDGEO  1.0     2001/07/30  Began Execution  Mon Aug 20 10:46:36 2001


 Input File Grid
    Grid Name              lake_erie
    X Origin                    0.
    Y Origin                    0.
    Cell Size, X-Direction      25000.
    Cell Size, Y-Direction      25000.
    # of Cells, X-Direction    18
    # of Cells, Y-Direction    12
    Boundary Thickness          0
    Projection Name         LAMBERT
    Projection Units        METERS
    1st Map Projection Parm    30.0000
    2nd Map Projection Parm    60.0000
    3rd Map Projection Parm   -84.0000
    X Center                 -84.0000
    Y Center                  41.0000

    # of Vertical Layers        1
    Vertical Coord Type
    Vertical Coord Top       **********
    Vertical Coord Layers    **********
```

****

---- Input File 10, /home/susick/test/GRIDGEO/lake_erie.vartab

Variable Parameters

Variable	Type	Method
FIPCODE	INT	DOMINANT
URW_FLAG	INT	AVERAGE
USGS_TYPE	INT	TOTAL
USGS_TOTAL	REAL	AVERAGE
FOREST	REAL	AVERAGE
AGRICULTURE	REAL	AVERAGE
TOTAL_AREA	REAL	AVERAGE
LAND	REAL	AVERAGE

---- Input File 10, /m3aux/emis/susick/smoketool/gridspec/lake_erie/sas/cy_ratios.dat
---- Input File 10, /m3aux/emis/susick/smoketool/gridspec/lake_erie/sas/cy_ratios.dat
---- Output File 31, /home/susick/test/GRIDGEO/lake_eriex.ioapix
Output File Level - 6 Set From Surrogate File
469 Records Written to Work File SRGWORK
82 Records Written to Work File OUT_APIX

URW_FLAG 026017 1 11 srgval,cytot= 1.28109E+08 1.28109E+08
URW_FLAG 026017 1 12 srgval,cytot= 2.86188E+08 4.14297E+08

URW_FLAG 026049 1 8 srgval,cytot= 1.53666E+08 1.53666E+08
URW_FLAG 026049 1 9 srgval,cytot= 4.82882E+08 6.36548E+08
URW_FLAG 026049 1 10 srgval,cytot= 2.72597E+08 9.09145E+08
URW_FLAG 026049 2 9 srgval,cytot= 4.10206E+08 1.31935E+09
URW_FLAG 026049 2 10 srgval,cytot= 2.46484E+08 1.56584E+09

URW_FLAG 026063 2 12 srgval,cytot= 8.79623E+07 8.79623E+07
URW_FLAG 026063 3 12 srgval,cytot= 3.03179E+08 3.91141E+08
URW_FLAG 026063 4 12 srgval,cytot= 2.70276E+08 6.61417E+08
URW_FLAG 026063 5 12 srgval,cytot= 7.47941E+07 7.36211E+08

---- lines dropped ----

USGS_TYPE 026017 1 11 srgval,celltot,ratio= 1.28109E+08 6.25000E+08 0.20497
USGS_TYPE 026017 1 12 srgval,celltot,ratio= 2.86188E+08 2.89572E+08 0.98831

USGS_TYPE 026049 1 8 srgval,celltot,ratio= 1.53666E+08 6.25000E+08 0.24587
USGS_TYPE 026049 1 9 srgval,celltot,ratio= 4.82882E+08 6.25000E+08 0.77261
USGS_TYPE 026049 1 10 srgval,celltot,ratio= 2.72597E+08 6.25000E+08 0.43615
USGS_TYPE 026049 2 9 srgval,celltot,ratio= 4.10206E+08 6.25000E+08 0.65633
USGS_TYPE 026049 2 10 srgval,celltot,ratio= 2.46484E+08 6.25000E+08 0.39438

USGS_TYPE 026063 2 12 srgval,celltot,ratio= 8.79623E+07 3.59210E+08 0.24488
USGS_TYPE 026063 3 12 srgval,celltot,ratio= 3.03179E+08 6.25000E+08 0.48509
USGS_TYPE 026063 4 12 srgval,celltot,ratio= 2.70276E+08 6.25000E+08 0.43244
USGS_TYPE 026063 5 12 srgval,celltot,ratio= 7.47941E+07 2.09261E+08 0.35742

---- lines dropped ----

```
    IOAPI Output File Written
---- Output File 20, /home/susick/test/GRIDGEO/lake_eriex.dat
    ASCII Output File Written
          9 Variables
         10 Header Records
         82 Data Records

GRIDGEO  1.0     2001/07/30  Ended Execution  Mon Aug 20 10:46:38 2001
STOP: NORMAL TERMINATION
```

Program Execution in a Models-3 Study

Gridgeo must be defined as a Models-3 program prior to being used in a study. The steps to do this are:

Click on the Program Manager icon from the main Models-3 screen. This will bring up the Program Manager.

Click on the New icon. This will bring up the properties window for a new program. Enter the following information in the window:

Program Name: gridgeo
Executable Full Path: $M3MSTOOL/bin/execProgram
Description Convert Gridded Data to Geographic Areas

Click on the Input Specs tab. This will bring up the input specification window for defining the program input files. The term NETCDF (below) is an acronym for Network Common Data Format. Define the following input specifications:

Logical Name	File Format	Dataset Type	Mandatory
GRIDDED	IOAPI NETCDF	*any type*	Y
SURROGATES	ASCII	demographic	Y
VAR_TABLE	ASCII	demographic	N

Click on the Output Specs tab. This will bring up the output specification window for defining the program output files. Define the following output specifications:

Logical Name	File Format	Dataset Type	Mandatory
OUT_API	IOAPI NETCDF	*any type*	N
OUT_APIX	ASCII	*any type*	N
OUT_ASCII	ASCII	*any type*	N

Click on the Env Vars tab. This will bring up a window for defining the program environment variables with their default values. Define the following environment variables:

Variable Name	Variable Value	Variable Description
LEVEL		Output File Level. (# of characters)
DEBUG	YES	Generate Debug Report? Y or N

Click on the Command Flags tab. This will bring up the command flag window for defining parameters that will be included on the execution command line for the program. Define the following command flag:

Flag Name	Flag Value	Flag Description
gridgeo		

Save the program by clicking on the Save icon. You can exit the Program Manager after the "program saved" message appears.

Once the program has been saved, it can be used in a study plan by annotating the program to a node. When annotating a node, you should:

Select gridgeo as the program in the Program Name field.

Enter a Node Name.

Change the environment variables to the settings desired.

Select the command line argument by clicking on Argument Name gridgeo. The line will be black with light letters when selected. If not selected, Models-3 will not include the parameter on the command line.

Annotate the input links to the node with the correct file specifications.

Annotate the output links from the node with the correct file specifications.

The node outputs will contain the following reports, which can be viewed by clicking on the node:

Menu Item	Report
View Latest Output	Messages written to standard output file
View Program Log	I/O API log file
View Program Report	Debug report file

APPENDIX B

Standalone I/O API Conversion Program

STANDALONE I/O API CONVERSION PROGRAM

The I/O API file conversion program is used by the Models-3 File Convertor to perform the following types of file conversions:

ASCII to IOAPI
IOAPI to IOAPI
IOAPI to ASCII
SAS to ASCII
ASCII to SAS

This Appendix describes how the program can be run in a standalone mode. It shows the environment variables used, sample scripts, and discusses the files used by the program. Environment variables define run options and file names used by the program. The environment variables used are:

MIDPROINFILE	Full pathname of input file.
MIDPROOUTFILE	Full pathname of output file.
TRANSACTION	Full pathname of file converter transaction file.
G_GRIDPATH	Full pathname of Models-3 grid description file (describes output file grid and vertical layers). Normally supplied by Models-3. Only needed for an ASCII to IOAPI conversion.
LOGFILE	Full pathname of I/O API log file.
MIDPROTEST	Number of records (or time steps if creating I/O API file) to write. Default is all records (or time steps).
DEBUG	Generate debug report? Valid values are YES, NO, Y, and N. The environment value is case-insensitive. Default is NO.
REPORT	Full pathname of debug report file. File is only generated when DEBUG is set to YES or Y. It shows the manner in which file, selection, and conversion specifications are built internally. It also displays the data mapped to I/O API variables for the initial records of an ASCII file.

M3FERRORFILE Full pathname of status. Only needed for NT. The status file tells Models-3 if the conversion was successful on the NT.

Conversion messages are written to the standard output (Unix or NT logs). Sample statements for running an ASCII to I/O API file conversion on Unix are shown below.

```
setenv BASDIR  /home/susick/file_converter
setenv MIDPROINFILE  $BASDIR/test/api_ascfix_no.dat
setenv MIDPROOUTFILE $BASDIR/test/asc_api_fix_no.ioapi
setenv TRANSACTION   $BASDIR/transactions/asc_api_fix_no.trans
setenv G_GRIDPATH    $BASDIR/transactions/m3grid_sf
setenv LOGFILE       $BASDIR/test/asc_api_fix_no.log
setenv DEBUG              yes
setenv REPORT        $BASDIR/test/asc_api_fix_no.report
$M3FBIN/M3TMapiconv.exe
```

The following statements run the same conversion on the NT:

```
set BASDIR=X:\file_converter
set MIDPROINFILE=%BASDIR%\test\api_ascfix_no.dat
set MIDPROOUTFILE=%BASDIR%\test_nt\asc_api_fix_no.ioapi
set TRANSACTION=%BASDIR%\transactions\asc_api_fix_no.trans
set G_GRIDPATH=%BASDIR%\transactions\m3grid_sf
set LOGFILE=%BASDIR%\test_nt\asc_api_fix_no.log
set DEBUG=yes
set REPORT=%BASDIR%\test_nt\asc_api_fix_no.report
set M3FERRORFILE=%BASDIR%\test_nt\asc_api_fix_no.status
%M3FBIN%\apiconv.exe
```

Both sets of statements assume that Models-3 environment variable M3FBIN has been defined. This environment variable defines the path of the Models-3 framework executable directory. In addition, the directories containing the I/O API and NetCDF libraries must be included in the system path on the NT.

The remainder of this document describes the simplified transaction file that can be used with the standalone program. The conversion program can use the transaction file generated by the Models-3 Graphical User Interface (GUI), but does not need all of the statements required by the GUI. Extensions have also been made for temperature unit conversions, and for readability in specifying file formats. The following conventions have been followed when describing statement formats:

User entered fields are shown in italics.

- Optional fields are enclosed in brackets.

The information entered on the statements is generally case-insensitive. Exceptions will be noted in the discussion.

Transaction File

The File Convertor transaction file specifies the manner in which a file will be converted. A transaction file consists of several sections. Each section consists of a section header record followed by detail records for the section. A transaction file can contain the following sections:

- Input file specification - Describes the input file. The section header describes the type of file, while the detail records describe the variables in the file. An I/O API input file only needs the section header. Any detail records for the file will be ignored, since the variable definitions will be taken from the I/O API file header information. An ASCII file specification must contain the detail records.

- Output file specification - Describes the output file. As in the input file specification, I/O API files do not need any detail records, whereas detail records are required for ASCII files. All input file variables (except character variables on an ASCII file) are copied to the output I/O API file when the detail records are omitted. When detail records are included, the variables defined in the specification are the only ones written to the output file.

- Selection criteria - Specifies data selection. The selection is based upon time, index, or data values.

- Conversion specification - Specifies the manner in which output file variables are computed from the input file variables. It can be used to rename variables or convert variables to different units.

The sections may be entered in any order for the conversion program. The only required sections are the input and output file specification. The selection criteria and conversion specification sections may be omitted. The section headers are free-format, space-delimited, and case-insensitive. The general formats of the section header statements are shown below:

#input type=f*iletype* [*ascii-file-parameters*]

#output type=f*iletype* [*ascii-file-parameters*]

#selection *selection-type*

#conversion

The following discussion describes each of the sections in more detail, and shows sample transactions. Any line in the transaction file whose first non-blank character is a pound sign (#) will be treated as a comment. In addition, any completely blank lines will be ignored.

Input File Specification

The input file specification describes the input file. The section header defines the type of file, while the detail record defines the variables on the file. The general format of the section header for each type of file is:

#input type=IOAPI
#input type=ASCII form=*format* [fldend=*delim*] [skip=*recskip*] [comment=*char*]

The header record parameters may be entered in any order. The extra parameters for an ASCII file are:

format Specifies the file format FIXED or FREE. May also specify 0 or 1, for FIXED or FREE respectively, to be consistent with the GUI.

delim Field delimiter for free-format files. Can be a character or one of the following special words: TAB, COMMA, or SPACE. Character values are case sensitive.

recskip Number of records to skip at beginning of file. Default is zero.

char Comment character (case sensitive). Input file records starting with this character are treated as comments.

The remaining records in the specification define the variables on the file. The general format of a variable definition is:

varname vartype start nchar format=*forma*t unitcat=*cat* unit=*units*

The keyword parameters may be entered in any order, but the positional parameters at the beginning of the statement must be entered in the order shown. The statement parameters are:

varname Variable name. Case sensitive.
vartype Variable type.
start Start position of variable on fixed format ASCII file record.
nchar Length of variable in characters.

format	Variable format.
cat	Units category of variable. May be enclosed in quotes. Case sensitive.
units	Units description of variable. May be enclosed in quotes. Case sensitive.

The valid variable types are:

Type	Description	ASCII File Format
MINT	Integer	Signed integer. Read or written using field length.
MFLOAT	Single Precision Real	Floating point number. Can use a format of the form F*w*.*d*, where *w* is the total field width and *d* is the number of digits to the right of the decimal point. The format is only used on output.
MDOUBLE	Double Precision Real	Floating point number. Can use a format of the form F*w*.*d*, where *w* is the total field width and *d* is the number of digits to the right of the decimal point. The format is only used on output.
MDATE	Date	Numeric date that consists of a 4-digit year, a 2-digit month, and a 2-digit day of month. Uses a date format that shows the relative position of year(y), month(m), and day(d) fields. Samples are ymd or yyyy/mm/dd. The samples specify that the date components are ordered year, month and day. The date form will be yyyy/mm/dd on ASCII file records. A date field must be at least ten characters long.
MTIME	Time	Time of Day. Contains a 24 hour time of the form hh:mm:ss. A time field must be at least eight characters long.

Type	Description	ASCII File Format
SASDATETIME	Date-Time	Combined date-time field (with an alphanumeric date) of the form ddmmmyyyy:hh:mm:ss. This is the SAS date-time format. A date-time field must be at least eighteen characters long.
DBLINT	Integer	Signed integer. Read or written using field length.
MCHAR	Character	Can be included in an ASCII file specification, however, the variable will not be copied to an I/O API file. Field will be blank on an output file.

The conversion program has a set of special variable names reserved for the I/O API file array indices. The names are case insensitive. These names are shown below:

Name	Type	Description
DATETIME	SASDATETIME	Time step date-time
DATE	MDATE	Time step date
TIME	MTIME	Time step time
LAY	MINT	Layer number
ROW	MINT	Row number
COL	MINT	Column number

To be consistent with the Models-3 GUI, the conversion program allows the specification to have either a DATE or TIME variable, instead of having a DATETIME variable or both the DATE and TIME variables. In those cases, the single variable (DATE or TIME) is treated as DATETIME.

Output File Specification

The output file specification describes the output file. The header record defines the type of file, while the detail record defines the variables on the file. The general format of the section header for each type of file is:

#output	type=IOAPI	
#output	type=ASCII	form=*format* [fldend=*delim*]

The header record parameters may be entered in any order. The extra parameters for an ASCII file are:

format	Specifies the file format FIXED or FREE. May also specify 0 or 1, for FIXED or FREE respectively, to be consistent with the GUI.
delim	Field delimiter for free-format files. Can be a character or one of the following special words: TAB, COMMA, or SPACE. Character values are case sensitive.

The remaining records in the specification define the variables on the file. The variable definitions are discussed in the Input File Specification section.

Selection Criteria

The formats of the selection header records are:

#selection	IOselect
#selection	nonIOselect

IOselect is normally used when creating an I/O API file (but it can also be used when creating an ASCII file). This specification can select a time period, a grid subset (with row and column ranges) or a set of vertical layers. The following detail statements can be used in an IOAPI selection specification to perform these selections:

start	*beg-date*	*beg-time*	Defaults to first time step on input file.
end	*end-date*	*end-time*	Defaults to last time step on input file.
row	*beg-row*	*end-row*	Defaults to all rows on input file.
column	*beg-column*	*end-column*	Defaults to all columns on input file.
layer	*layer1 layer2 ... layern*		Defaults to all layers on input file.

NonIOselect can only be used when creating an ASCII file. This specification can be used to select using the file indices or the contents of any variable on the file. The detail records define a set of “ANDed” logical conditions that define a select or not-select condition for array entries. The general format of these expressions are:

varname	*operator*	*value*
varname	IN	*{value1,value2,...,valuen}*
varname	NOT IN	*{value1,value2,...,valuen}*

varname NOTIN *{value1,value2,...,valuen}*

where *varname* is the name of an input file variable. The conversion program will use a variable that exactly matches *varname*, if one exists on the input file. Otherwise it will use the first variable it finds that has the same letters, regardless of case.

The logical operators (*operator*) that may be entered in an expression are:

=	equal to
!=	not equal to
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

IN selects an array entry if the variable value for that entry is in the value list. NOTIN selects the entry if the value is not in the value list. IN and NOTIN should only be used with integer, date-time, date, or time variables, since they perform exact matches on the values in the list.

Dates are entered as yyyy/mm/dd in the preceding statements. Times are entered as hh:mm:ss, hh:mm, or hh. Missing seconds and minutes default to zero. All date and time components except year can be entered as one digit, dropping the leading zero. Date-time variables entered in a NonIOselect expression are a combination of the date and time forms entered above (*date:time*). Some valid date-time values are:

1995/7/11:10:0:0	July 11, 1995 10am
1995/08/01:15:30	August 1, 1995 3:30pm

Conversion Specification

The conversion specification specifies the manner in which output file variables are computed from input file variables. It can be used to rename variables or convert variables to different units. The specification consists of a header record followed by detail records that specify the conversion for individual variables. The format of the header record is:

#conversion

The format of the detail records is:

in *out* [*factor*]

where the parameters are:

in Name of the input file variable.
out Name of the output file variable.
factor Conversion factor. Default is one.

If the conversion factor is entered, it must be one of the following temperature conversion types or a positive number:

T_CTOF Degrees Celsius to Fahrenheit
T_CTOK Degrees Celsius to Kelvin
T_FTOC Degrees Fahrenheit to Celsius
T_FTOK Degrees Fahrenheit to Kelvin
T_KTOC Degrees Kelvin to Celsius
T_KTOF Degrees Kelvin to Fahrenheit

The temperature conversion types are case-insensitive. When a number is entered, the variable is computed using the following equation:

$$out = in \times factor$$

Conversion records only have to be entered if:

- The conversion factor is not one.
- The output variable name does not match the input variable name. The conversion program matches variables on the files using a two-pass search. The first pass matches variable whose names exactly match. The second pass matches variables having the same name, regardless of case.

Sample Transaction Files

IOAPI to IOAPI Conversion	Full copy of IOAPI file

```
#input type=IOAPI
#output type=IOAPI
```

IOAPI to IOAPI Conversion	Select times 10:00:00 to 15:00:00 on 8AUG1995 Select rows 2 to 6 Select columns 2 to 5 Copy all variables

```
#input type=IOAPI
#selection IOselect
start  1995/08/01  10:00:00
end    1995/08/01  15
row    2 6
column 2 5
#output type=IOAPI
```

IOAPI to IOAPI Conversion	Select times 10:00:00 to 15:00:00 on 11JUL1995 Select variables QC, QR, TA, QV, and PRES

```
#input type=IOAPI
#selection IOselect
start  1995/07/11  10:00:00
end    1995/07/11  15
#output type=IOAPI
QC
QR
TA
QV
PRES
```

IOAPI to ASCII Conversion	Fixed format output file Drop times 10:00:00,12:00:00, and 14:00:00 for 8AUG1995 Drop rows 1,3, 5, and 7 Select columns 2, 4, and 6

```
#input type=IOAPI
#output type=ASCII form=FIXED
time SASDATETIME   1 20
col  MINT    21  5
row  MINT    26  5
ALD  MFLOAT  31 10  format=F10.7
CO   MFLOAT  41 10  format=F10.4
HC8  MFLOAT  51 10  format=F10.4
ETH  MFLOAT  61 10  format=F10.7
```

```
NO   MFLOAT  71 10  unitcat=emissions unit=GramPerSecond

#selection  nonIOselect
datetime notin {1995/08/01:10:00:00, 1995/08/01:12:00, 1995/08/01:14}
row not in {1,3,5,7}
col in {2,4,6}
```

IOAPI to ASCII Conversion

Fixed format output file
Convert temperatures to different units
Select ambient temperature > 295 deg K

```
#input type=IOAPI
#selection nonioselect
ta > 295
#Conversion
  ta  takC  t_ktoc
  ta  takF  t_ktoF
  tac tacf  t_ctof
  tac tack  t_ctok
  taf tafc  t_ftoc
  taf tafk  t_ftok
#output type=ascii form=0
lay   mint    1 3
row   mint    4 3
col   mint    7 3
TA    MFLOAT  11 10 unitcat= units= format=f10.3
TACK  MFLOAT  21 10 unitcat= units= format=f10.3
TAFK  MFLOAT  31 10 unitcat= units= format=f10.3
TAC   MFLOAT  41 10 unitcat= units= format=f10.3
TAFC  MFLOAT  51 10 unitcat= units= format=f10.3
TAKC  MFLOAT  61 10 unitcat= units= format=f10.3
TAF   MFLOAT  71 10 unitcat= units= format=f10.3
TACF  MFLOAT  81 10 unitcat= units= format=f10.3
TAKF  MFLOAT  91 10 unitcat= units= format=f10.3
```

ASCII to IOAPI Conversion

Free format input file, comma-delimited

```
#output type=IOAPI

#input type=ASCII form=free  fldend=comma
date MDATE    1 10  format=yyyy/mm/dd
time MTIME   11 10
col  MINT    21  5
```

```
row  MINT    26  5
ALD  MFLOAT  31 10  format=F10.7 unitcat=emissions unit=g/s
CO   MFLOAT  41 10  format=F10.4 unitcat=emissions unit=g/s
HC8  MFLOAT  51 10  format=F10.4 unitcat=emissions unit=g/s
ETH  MFLOAT  61 10  format=F10.7 unitcat=emissions unit=g/s
```

Program Execution in a Models-3 Study

A Models-3 program must be defined for each type of conversion. Three programs will be needed before performing conversions in a study plan:

asci_to_api	ASCII to IOAPI conversion program
api_to_api	IOAPI to IOAPI conversion program
api_to_asci	IOAPI to ASCII conversion program

The primary reason for having three programs is that each conversion type has different file formats for the conversion input and output files (MIDPROINFILE and MIDPROOUTFILE). The steps below show how to define the ASCII to IOAPI conversion program. The other two programs can be defined in a similar manner changing names, descriptions, and file formats as appropriate.

- Click on the **Program Manager** icon from the main Models-3 screen. This will bring up the Program Manager.

- Click on the **New** icon. This will bring up the properties window for a new program. Enter the following information in the window:

  Program Name: asci_to_api
  Executable Full Path: $M3FBIN/apiconv.exe
  Description: Convert ASCII to IOAPI

- Click on the Input Specs tab. This will bring up the input specification window for defining the program input files. Define the following input specifications:

Logical Name	File Format	Dataset Type	Mandatory
MIDPROINFILE	ASCII	*any type*	Y

- Click on the Output Specs tab. This will bring up the output specification window for defining the program output files. Define the following output specifications:

Logical Name	File Format	Dataset Type	Mandatory
MIDPROOUTFILE	IOAPI NETCDF	*any type*	Y
LOGFILE	ASCII	*any type*	Y

- Click on the Env Vars tab. This will bring up the environment variable window for defining the program environment variables with their default values. Define the following environment variables:

Variable Name	Variable Value	Variable Description
MIDPROTEST		Number of Time Steps to Write
DEBUG	YES	Generate Debug Report? Y or N
REPORT	$STUDY_PATH /program.lst	Debug report file name

  MIDPROTEST will be the number of records to write when creating an ASCII output file.

- Save the program by clicking on the **Save** icon. You can exit the Program Manager after the "program saved" message appears.

Once the program has been saved, it can be used in a study plan by including the program when annotating a node. At this time you should:

- Select appropriate program for the Program Name field.
- Enter a Node Name.
- Change the environment variables to the settings desired.
- Annotate the input links to the node with the correct file specifications.
- Annotate the output links from the node with the correct file specifications.

The node outputs will contain the following reports, which can be viewed by clicking on the node:

Menu Item	Report
View Latest Output	Messages written to standard output file

View Program Log	I/O API log file
View Program Report	Debug report file

APPENDIX C

Preparation of Spatial Grid Surrogates in SMOKE Tool

Grid Spatial Surrogate Preparation

The primary purpose of the grid processor is to generate spatial surrogates for SMOKE horizontal gridding. The grid processor is executed through nodes in a study plan. The underlying functions that the grid processor executes are modified versions of the Statistical Analysis System (SAS) and ARC/INFO gridding programs copied from the

MEPPS Emission Processor (EMPRO). The grid processor performs the following functions:

- Define Grid. Define creates the grid workspace, receiving the grid description file from the Study Planner. The program creates the grid directories, imports the grid description, and generates the base Geographic Information System (GIS) coverages in the workspace. The grid description is read from file G_GRIDPATH (supplied from the study planner).

- Generate Coverages. This action generates GIS coverages in the workspace. The user selects the coverages to generate through environment variables. The names of the environment variables are based upon the GIS coverages defined to the SMOKE Tool.

- Generate Surrogates. This action computes spatial surrogates for the grid from the GIS coverages generated for the grid. The user selects the coverages to include in the surrogate calculation through environment variables. The program computes all spatial surrogates defined for each coverage selected.

This grid processor will create the grid directories in a user-specified location. The location is specified through the following environment variables:

EMS_HOME Workspace Path. The base directory under which the grid information will be stored. Supplied by user.

HGRIDNAME Models-3 Grid Name. The grid name is used to define subdirectories under the workspace path for the grid. Supplied by study planner.

All data for the grid, created by the Define Grid and Generate Coverages functions, will be stored in subdirectories under directory $EMS_HOME/gridspec/$HGRIDNAME. The user will specify the locations of the spatial surrogate files, created by the Generate Surrogate function, in plan output file specifications.

The user defines the GIS coverages available to SMOKETOOL through the following environment variables:

GISDB Name of the GIS data base directory. This directory contains the GIS coverages and related data files for the surrogate processing.

COVER_DEF	Name of the file containing the coverage definition table. This table defines the GIS coverages available for surrogate processing.

The GIS coverages released with Models-3 are:

COUNTY	County areas	
CENSUS	Census tract/block group areas	(US only)
FHAROAD	FHA road lengths	(US only)
AGRICULTURE	Agricultural areas	(US only)
AIRPORTS	Airport locations	
PORTS	Port locations	
RAILROADS	Railroad lines	
LAND_WATER	Land-water areas	
URBAN-RURAL	Urban-rural areas	
ROADS	Road lengths	
FOREST	Forest areas	
TIGER	TIGER/Line roads	(US only)
WATERSHED	Watersheds	(US only, non-county)

To select a coverage in the coverages step, set the environment variable for the coverage to YES or Y. The value may contain upper or lower case letters. Sample environment variable settings that will cause coverages to be processed are:

```
PORTS=Y
AIRPORTS=YES
RAILROADS=yes
WATERSHED=y
```

Selecting coverages for the surrogates step is a bit more complicated, since the spatial surrogates for non-county coverages are written to individual output files and all other surrogates (which are county-based) are written to a single file (specified in environment variable SURROGATES). To select a non-county coverage, enter the output file name in the coverage environment variable. Set the coverage environment variable to YES or Y for the other coverages. Sample environment variable settings that will cause coverages to be processed are:

```
PORTS=Y
AIRPORTS=YES
RAILROADS=yes
WATERSHED=/home/user/watershed.ratios
```

The surrogates step also uses the following environment variables which supply the names of input files:

FEATURE_SRG Name of the Feature-Surrogate Table. This table defines the manner in which spatial surrogates are computed from the coverage features.

CENSUS_DATA Name of the Census Data File. This file contains housing and population data associated with the CENSUS coverage. May also contain state or county level data.

FEAT_STCY_FRAC Name of the Census State-County Fraction table. This file contains fractions to allocate state level census data to the counties within the states.

The follow sections describe the various directories, GIS coverages, and files used in the spatial surrogate processor.

Grid Directories

The grid master directory is $EMS_HOME/gridspec/$EMS_GRID. The subdirectories created are shown in the following table:

Subdirectory	Type of Data	Environment Variable	SAS Reference
common/sas	Ungridded Emissions	EMS EMS_CVRT	library EMS file EMS_CVRT
common/gis	Ungridded GIS	EMSG	file EMSG
sas	Gridded Emissions	EMS_GRD EMSF_GRD	library EMS_GRD file EMSF_GRD
gis	Gridded GIS	EMSG_GRD	file EMSG_GRD

The first two directories contain data for the grid's geographic area that have not been processed for a grid. The EMSG directory contains GIS coverages that have not been overlaid with a grid.

The gridded data directories contain data that have been processed for the grid. EMS_GRD contains the grid description and gridded data files. The EMSG_GRD directory contains GIS coverages that have been overlaid with the grid.

Coverage Definition Table

The coverage definition table defines the GIS coverages available to the SMOKE Tool. The table is a free-format space-delimited ASCII file containing the following information on each record:

Field	Name	Type	Description
1	COVNAME	Char	SMOKE Tool coverage name. This name is the identifier used for the coverage in the SMOKE Tool. It provides the name of the environment variable used to select the coverage in processing, and is the name used to reference the coverage in SMOKE Tool processing
2	COUNTRY	Char	Country name for GIS coverage
3	COVTYPE	Int	Coverage type (poly, line, point).
4	COVERAGE	Real	Coverage name in the GISDB directory. This name defines the location of the coverage to ARC
5	COVLEV	Char	Coverage level. “County” specifies a county level coverage. Any other value is the item name of the ID variable in a non-county coverage. Default is county.

A SMOKE Tool coverage can contain one or more GIS coverages. In the simplest instance, the Tool coverage contains a single GIS coverage that contains all countries. In that case, the coverage definition table contains a single record with a country name of “ALL”. The more common instance is that there are individual GIS coverages for each country, or regions in the countries. In that case, the table will contain one or more entry for each country. The entries will each contain a country name (US, CANADA, MEXICO). The SMOKE Tool coverage generation will treat the set of GIS coverages under a coverage name as a single coverage.

A sample table is shown below:

```
COUNTY       ALL      poly   na_geo      county
CENSUS       US       poly   us_bg90     county
FHAROAD      US       line   fhards_48   county
AGRICULTURE  US       poly   us-ag       county
AIRPORTS     US       point  us-ap       county
AIRPORTS     CANADA   point  can-ap      county
PORTS        US       point  us-port     county
PORTS        CANADA   point  can-port    county
RAILROADS    US       line   us-rr       county
```

```
RAILROADS    CANADA   line   can-rr      county
LAND_WATER   US       poly   us-lwcty    county
LAND_WATER   CANADA   poly   can-lwcty   county
URBAN_RURAL  US       poly   us-urcty    county
URBAN_RURAL  CANADA   poly   can-urcty   county
ROADS        US       line   us-rd       county
ROADS        CANADA   line   can-rd      county
FOREST       US       poly   us-for      county
FOREST       CANADA   poly   can-for     county
WATERSHED    US       poly   us-ws1      huc
WATERSHED    US       poly   us-ws2      huc
WATERSHED    US       poly   us-ws3      huc
WATERSHED    US       poly   us-ws4      huc
```

The following restrictions apply to the GIS coverages grouped under a coverage name:

The coverages must have the same coverage type (poly, line, or point).

The coverages must be discrete, with no IDs spanning coverages.

County level coverages must contain entire states. This is required for computing state level spatial surrogates.

The coverages do not have to define the entire country, but must cover the grid domain.

GIS Coverages

The GIS data base (GISDB) directory contains the GIS coverages used by the SMOKE Tool. The GISDB directory contains two types of coverages:

Coverages from the original SMOKE Tool spatial surrogate processor. These include the COUNTY, CENSUS, and FHAROAD coverages. TIGER road coverages are also included, but are located in a separate directory (TIGERDB).

County level universal coverages.

Non-county universal coverages.

All user added coverages must be universal coverages. Universal coverages must have the following projection parameters

Projection	Geographic
Units	DD
Datum	NAD83

Spheroid GRS1980

The only required field in a non-county universal coverage is the data item specified as the ID variable in the coverage definition file. County level universal coverages must contain the following data items:

STATE	Country/state code. For US coverages this is the 2-digit FIPS state code. For other countries, this is a 3-digit field with the country code in the first digit and the state (or province) code in the second and third codes.
COUNTY	County code. This is a 3-digit field for all coverages. The field contains the FIPS county code for US coverages.
STIDCYID	Combined country/state/county code. This field may be a redefinition of the STATE and COUNTY fields. For US coverages this may be a 5-digit field. For other countries it must be a 6-digit integer field.
COV_CODE	Feature code. Classifies features within the coverage. May be zero if the coverage only has one feature. In the CENSUS coverage, this field contains the census id. For US coverages this will either be a census tract or block group ID. The feature codes for the CENSUS coverage are defined in the CENSUS_DATA file included with the coverage, which contains the housing and population data associated with the coverage. The field length can be one to sixteen digits, based upon the coverage.

The feature codes defined in the original universal coverages are shown below:

Coverage	Code	Description
AGRICULTURE	1	Agriculture Area
	2	Non-Agriculture Area
AIRPORTS	0	Airport Location
PORTS	0	Port Location
RAILROADS	0	Railroad
LAND_WATER	1	Water Area
	2	Island Area
	3	Mainland Area

URBAN_RURAL	1	Urban Area
	2	Rural Area
ROADS	01	Unclassified Urban Road
	02	Unclassified Rural Road
	11	Urban Primary Road
	12	Rural Primary Road
	21	Urban Secondary Road
	22	Rural Secondary Road
FOREST	1	Forest Area
	2	Non-forest Area

Gridded Data Files

The coverages step generates gridded data files in the EMS_GRD directory for each coverage processed. The gridded data files are free-form comma-delimited ASCII files. The coverage step generates two files for each county level universal coverage:

covname.cy County level gridded data
covname.st State level gridded data

The fields on each file record are:

Field	Name	Type	Description
1	STID	Int	Country/StateCode
2	CYID	Int	County code (zero for state level data)
3	FEATURE	Char	Feature Code
4	COL	Int	Grid column number (zero if out of grid)
5	ROW	Int	Grid row number (zero if out of grid
6	GVAL	Real	Size of feature in county (or state)-grid cell intersection
7	CYVAL	Real	Size of feature in county (or state)

The coverages step only generates one gridded data file(census.cy) for the CENSUS coverage. It is also a free-form comma-delimited ASCII file. In place of the feature

code, the file contains the census id, which in the US coverage is a census block group ID.

The coverages step also generates a single file (*covname*.geo) for each non-county universal coverage processed. The fields on each file record are:

Field	Name	Type	Description
1	COVID	Char	Coverage ID
2	COL	Int	Grid column number (zero if out of grid)
3	ROW	Int	Grid row number (zero if out of grid
4	GVAL	Real	Size of feature in county (or state)-grid cell intersection
5	CYVAL	Real	Size of feature in county (or state)

Feature-Surrogate Table

The Feature-Surrogate table defines the manner in which spatial surrogates are computed from the features in county level coverages. The table is a free-format space-delimited ASCII file containing the following information on each record:

Field	Name	Type	Description
1	COVNAME	Char	SMOKE Tool coverage name. This name is the identifier used for the coverage in the SMOKE Tool. It provides the name of the environment variable used to select the coverage in processing, and is the name used to reference the coverage in SMOKE Tool processing. The coverage must be a county level universal coverage or one of the original SMOKE Tool coverages.
2	FEATURE	Char	Coverage feature code
3	SSC	Int	Spatial surrogate code
4	FACTOR	Real	Fraction of feature added to spatial surrogate (default is one)

A feature may be included with more than one spatial surrogate, either entirely or partially. A sample table is shown below:

```
#  Coverage Feature Code to Spatial Surrogate Code Table
#  covname     feature  ssc  factor
   AGRICULTURE  1        1      1     Agriculture (US)
   CENSUS       AGAREA   1      1     Agriculture (Canada)
   AIRPORTS     0        2      1     Airports
   LAND_WATER   2        3      1     Land Area (Island)
   LAND_WATER   3        3      1           (Mainland)
   CENSUS       TOTHU    4      1     Total Housing Units
   ROADS        11       7      1     Major Highways (Urban Primary Roads)
   ROADS        12       7      1           (Rural Primary Roads)
   CENSUS       TOTPOP   8      1     Total Population
   PORTS        0        9      1     Ports
   RAILROADS    0        10     1     Railroads
   LAND_WATER   1        11     1     Water Area
   URBAN_RURAL  2        12     1     Rural Area
   URBAN_RURAL  1        13     1     Urban Area
   FOREST       1        14     1     Forest Area
   ROADS        11       15     1     Urban Primary Roads
   ROADS        12       16     1     Rural Primary Roads
   ROADS        21       17     1     Urban Secondary Roads
   ROADS        22       18     1     Rural Secondary Roads
   CENSUS       URBPOP   19     1     Urban Population
   CENSUS       RURPOP   20     1     Rural Population
```

Except for the CENSUS coverage, each table entry refers to a feature code defined in the coverage, and returned on the gridded data file generated for the coverage. The spatial surrogates are computed using the gridded areas, lengths or point counts returned for the coverage features.

The CENSUS entries refer to feature codes defined on the census data file. CENSUS surrogates are computed by applying census data to gridded areas returned from the CENSUS and COUNTY coverages. Census id level data is applied the gridded CENSUS data. County level data area is applied to the gridded COUNTY areas. State level data is first apportioned to counties and then applied to the gridded COUNTY areas.

Census Data File

The census data file contains data that can be used in conjunction with spatial data at the census id or county level to compute spatial surrogates. The file is a free-form space-delimited ASCII file with the following information on each record:

Field	Name	Type	Description
1	STID	Int	Country/State/County Code
2	CYID	Int	County code (zero indicates missing)
3	CENSUSID	Char	Census ID (zero indicates missing)
4	FEATURE	Char	Feature Code
5	VALUE	Real	Feature Value

The data can be entered at the state, county or census id level. Census id data has a stid, cyid and censusid. County level data has the stid and cyid, but no censusid. State level data only has the stid.

The feature codes in the current census data files are:

TOTHU	Total housing units	
TOTPOP	Total population	
RURPOP	Rural population	
URBPOP	Urban population	
AGAREA	Agricultural area	(used for Canada)

Sample US and Canadian data is shown below. The US data is at the census block group level. The Canadian data is at the province level.

```
1 1 02011  RURPOP 65
1 1 02011  TOTHU  370
1 1 02011  TOTPOP 1105
1 1 02011  URBPOP 1040
1 1 02012  RURPOP 50
1 1 02012  TOTHU  301
1 1 02012  TOTPOP 756
1 1 02012  URBPOP 706
1 1 02021  RURPOP 0
1 1 02021  TOTHU  273
1 1 02021  TOTPOP 783
1 1 02021  URBPOP 783
1 1 02022  RURPOP 0
1 1 02022  TOTHU  419
1 1 02022  TOTPOP 1194
1 1 02022  URBPOP 1194
   .
   .
   .
160 0  0 RURPOP 32085
160 0  0 TOTHU  11813
160 0  0 TOTPOP 32085
```

```
160 0  0 URBPOP 0
161 0  0 RURPOP 1218376
161 0  0 TOTHU  306468
161 0  0 TOTPOP 1218376
161 0  0 URBPOP 0
111  0  0 AGAREA 141021.2778
112  0  0 AGAREA 12369.2827
113  0  0 AGAREA 684.8532
124  0  0 AGAREA 27594.2432
135  0  0 AGAREA 90992.9167
146  0  0 AGAREA 103307.607
147  0  0 AGAREA 1214942.9943
148  0  0 AGAREA 10979.4997
159  0  0 AGAREA 62492.9607
```

Census State-County Fraction Table

This table supplies factors that allocate state level census data to counties. The file is a free-form space-delimited ASCII file with the following information on each record:

Field	Name	Type	Description
1	COUNTY	Int	Country/State/County Code
2	FEATURE	Char	Feature Code
3	FRAC	Real	Fraction of state feature value in county

The following lines allocate Newfoundland (province 110) total population to counties for the spatial surrogate calculation:

```
110001 TOTPOP  0.44540757
110002 TOTPOP  0.05161995
110003 TOTPOP  0.04263404
110004 TOTPOP  0.04519323
110005 TOTPOP  0.07971190
110006 TOTPOP  0.07077839
110007 TOTPOP  0.07594072
110008 TOTPOP  0.09126545
110009 TOTPOP  0.04401675
110010 TOTPOP  0.05343200
```

County Level Spatial Surrogate File

The county level spatial surrogate file is a free-form, space-delimited ASCII file. It consists of a header record followed by one or more detail records. The header record

contains a description of the grid. The detail records contains spatial surrogate data for the grid. The file contains spatial records for both states and counties. The format of the header record is:

#GRID param1 param2 ... param15

The header record parameters are show below:

Parameter	Name	Type	Description
1	GDNAME	Char	Grid name (up to 16 characters)
2	XORIG	Real	X origin (projection units)
3	YORIG	Real	Y origin (projection units)
4	XCELL	Real	Cell size, X direction (projection units)
5	YCELL	Real	Cell size, Y direction (projection units)
6	NCOLS	Int	Number of cells in X direction
7	NROWS	Int	Number of cells in Y direction
8	NTHIK	Int	Boundary thickness (number of cells)
9	PROJTYPE	Char	Project type (UTM, LAMBERT, LAT-LON,...)
10	PROJUNIT	Char	Projection units (DEGREES or METERS)
11	P-ALP	Real	First map projection description parameter
12	P_BET	Real	Second map projection description parameter
13	P_GAM	Real	Third map projection description parameter
14	P_XCENT	Real	Center of coordinate system (degrees longitude)
15	P_YCENT	Real	Center of coordinate system (degrees latitude)

The format of the spatial surrogate records is shown below:

Field	Name	Type	Description
1	SSC	Int	Spatial Surrogate Code
2	COUNTY	Int	Country/State/County Code

Field	Name	Type	Description
3	COL	Int	Grid Column Number (x-axis)
4	ROW	Int	Grid Row Number (y-axis)
5	RATIO	Real	Spatial Surrogate Ratio (fraction of county value in grid cell)
6	VALUE	Real	Surrogate Value for county-grid cell intersection

There is only one county level surrogates file generated in a surrogates step. The file name is defined in the SURROGATES environment variable.

Non-County Spatial Surrogate Files

Non-county spatial surrogate files are each generated from a single non-county universal coverage. Each is a free-form, space-delimited ASCII file, that consists of a header record followed by one or more detail records. The header record contains a description of the grid, and is in the same format as the county level surrogate file header. The format of the spatial detail records is shown below:

Field	Name	Type	Description
1	COVID	Char	Coverage ID. This will be the coverage item defined as the ID in the coverage definition table.
2	COL	Int	Grid Column Number (x-axis)
3	ROW	Int	Grid Row Number (y-axis)
4	RATIO	Real	Spatial Surrogate Ratio (fraction of county value in grid cell)
5	VALUE	Real	Surrogate Value for county-grid cell intersection

Multiple non-county surrogate files may be generated in a surrogates step (one for each coverage selected). The coverage environment variable will contain the name of the file containing the surrogates computed for the coverage. For example, the file defined in environment variable WATERSHED will contain the spatial surrogates computed from the WATERSHED coverage.

APPENDIX D

Modifications to Allow Adding and Changing Spatial Surrogate Coverages in SMOKE Tool

Changing and Adding Spatial Surrogate Coverages in SMOKE Tool

The SAS portion of the SMOKETOOL spatial surrogate processor has been modified to allow processing of universal non-county coverages (watershed coverages will be of this type). Basically SAS passes a coverages file to ARC that specifies the coverages to process. The ARC programs then create a set of ASCII output files for each coverage that contain gridded data.

The first modification was to add a fourth field to file (coverages) passed to ARC. This file tells ARC which GIS coverages to process. The new record format is:

covname, covtype, coverage, codename

The first three fields remain unchanged. They contain the following information:

covname	SMOKETOOL coverage name
covtype	coverage type (point, line, poly)
coverage	GIS coverage name (pathname of coverage in GISDB directory)

The fourth field (codename) is new. It contains a description of the ID returned in the coverage output file. It will be "county" for the existing types of coverages (county, census, universal-county, ...). It will contain the item name of the geographic area ID for universal non-county coverages (like “huc” for a watershed coverage). With the code changes, the ARC programs will always receive a value in the code name field.

The output file that ARC generates from a universal non-county coverage will be *covname*.geo, instead of the *covname*.cy and *covname*.st files generated for a universal county-level coverage. The formats for the *covname*.geo files will be:

code, col, row, cellsize, totsize	(for line and poly coverages)
code, col, row	(for point coverages)

Instead of the following formats generated for the *covname*.cy and *covname*.st files:

stid, cyid, feature, col, row, cellsize, totsize	(for line and poly coverages)
stid, cyid, feature, col, row	(for point coverages)

The record fields shown above are:

code	ID for geographic area
col	Grid column area
row	Grid row number
cellsize	Size of geographic area-cell intersection
totsize	Total size of geographic area
stid	State code
cyid	County code
feature	Coverage feature code

The SAS and ARC programs for the SMOKETOOL Spatial Surrogate Processor are stored in directory $M3MSTOOL/bin/grid. The important ARC programs are:

exnatl.aml	Controlling AML program
point_grid.aml	AML program for universal county-level point coverages

univ_grid.aml	AML program for universal county-level line and poly coverages

The SAS program that executes exnatl.aml is exnatl.sas. Exnatl.aml will have to be changed to allow for the universal non-county coverages. The other two aml programs can serve as patterns for equivalent programs to process the non-county coverages.

The statements in exnatl.sas that read the coverages file are:

```
&s rfu [open coverages rstat -read]
&s line [read %rfu% rst]
&do &while %rst% = 0

 &s evar [extract 1 [unquote %line%]]
 &s feat [extract 2 [unquote %line%]]
 &s covname [extract 3 [unquote %line%]]
```

They currently extract three fields from each record. They will have to be changed to extract a new fourth field. They are followed by statements that decide which programs to execute for the coverage:

```
/* If the coverages are TIGER, US Census, or US Roads then
/* a separate script will be run to process them, all other
/* coverage will be processed by the same script

  &if %evar% eq TIGER or %evar% eq FHAROAD or %evar% eq COUNTY
  &then
  &do
   &if %evar% = TIGER &then
     &r mvextig offnet
/*   &if %evar% = CENSUS &then
/*     &r census_grid
   &if %evar% = FHAROAD &then
     &r fhard_grid
   &if %evar% = COUNTY &then
     &r cntyrd_grid
  &end

/* Running scripts to process all other coverages
  &else
  &do
    &type *****Processing [translate %evar%]*****
    &if %feat% = point &then
      &r point_grid %covname% %evar%
    &else
      &r univ_grid %covname% %feat% %evar%
```

```
&end
```

These statements will also have to be changed. The changes will be in the else block that processes all other coverages. Those are the statements for the universal county-level coverages.

Note that ARC programs that create the *covname*e.geo file must write the file in append mode, since there may be multiple GIS coverages contributing to the file. point_grid.aml and univ_grid.aml already do this. SAS program exnatl.sas deletes the files to be generated so that the ARC programs do not have to worry about initializing the files.